

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

2  
3  
4  
5

## 7

8  
9  
10  
11  
12

13

14

16  
17  
18  
19  
20  
21  
22

24

25

1 requirements necessary to support execution of the development project as noted  
2 in the “Summary” section. Put another way, the various claimed embodiments are  
3 directed to, in some way, reduce source filter invocation in a development project.  
4 Hence, the title is not misdescriptive at all. Rather, the title adequately  
5 characterizes various claimed embodiments.

6 Applicant has amended the specification to update the “Related  
7 Applications” section with the Applicant Serial numbers of the related  
8 applications.

### 9 10 **§101 Rejections**

11 Claims 1-12 and 32-43 stand rejected under 35 U.S.C. § 101 as being  
12 directed to non-statutory subject matter. Specifically, the Office argues that the  
13 claimed subject matter is software and that the software is not embodied on a  
14 computer readable medium.

15 Applicant has amended the subject claims to recite that the claimed subject  
16 matter resides on a computer readable media. Accordingly, Applicant respectfully  
17 traverses the Office’s rejection.

### 18 19 **§102/103 Rejections**

20 Claims 1 and 3-14 stand rejected under 35 U.S.C. §102(e) as being  
21 anticipated by U.S. Patent No. 6,535,920 to Parry et al. (hereinafter “Parry”).

22 Claims 2 and 37 stand rejected under 35 U.S.C. §103(a) as being obvious  
23 over Parry in view of Official Notice taken by the Office.  
24  
25

1 Claims 32-36 and 38-48 stand rejected under 35 U.S.C. §103(a) as being  
2 obvious over Parry in view of U.S. Patent No. 6,442,658 to Hunt et al. (hereinafter  
3 “Hunt”).

4 Before undertaking a discussion of the substance of the Office’s rejections,  
5 the following discussion of Applicant’s disclosure is provided in an attempt to  
6 assist the Office in appreciating the patentable distinctions between the claimed  
7 embodiments and the material cited and applied by the Office.

### 8 9 **Applicant’s Disclosure**

10 Perhaps a good place to begin to appreciate the various claimed  
11 embodiments is with a problem outlined in the “Background” section of the  
12 present application. Specifically, the “Background” section instructs that  
13 construction and implementation of filter graphs is computationally intensive and  
14 expensive in terms of memory usage. See, e.g. page 3, lines 12-21. As noted,  
15 even the most simple of filter graphs require an abundance of memory to facilitate  
16 the copy operations required to move data between filters. Thus, complex filter  
17 graphs can become unwieldy, due in part to the linear nature of conventional  
18 development system architecture. Moreover, it is to be appreciated that the filter  
19 graphs themselves consume memory resources, thereby compounding the issue  
20 introduced above.

21 Thus, the various claimed embodiments are directed to systems which  
22 reduce the computational and memory resources required to support even the most  
23 complex of multimedia projects.  
24  
25

1 Consider now Applicant's Figs. 40-43 and the following discussion. The  
2 opening and processing of media represents consumption of memory and  
3 processing resources. Thus, performance improvements may be achieved by  
4 reducing the number of times a source is accessed. In the discussion that follows,  
5 a method is presented, in accordance with one embodiment, that serves to reduce  
6 the number of times a source is accessed, e.g., a method of source combining. It is  
7 to be appreciated, however, that the following is but one example implementation  
8 of the broader concept of reducing the number of times a source need be accessed  
9 during execution of a development project.

10 **Fig. 41** illustrates an example method of generating a filter graph, in  
11 accordance with one embodiment. As shown, method 4000 begins with block  
12 4002, wherein render engine 222 (Fig. 3) receives an indication to generate a  
13 development project. According to one implementation, render engine 222  
14 receives the indication from a higher-level application 216, e.g., media processing  
15 application 224, to assist a user in generating a processing project (e.g., a media  
16 processing project).

17 In block 4004, render engine 222 identifies the number and nature of the  
18 media sources within the user-defined processing project, in preparation for  
19 generating a filter graph representation of the processing project. For each of the  
20 identified sources, render engine 222 determines the necessary transform filters  
21 306 required to pre-process the source (i.e., source chain), preparing the  
22 processing chain for presentation to the matrix switch filter 308 and one or more  
23 transition/effect filters 306. Unlike conventional implementations, which would  
24 proceed to generate the entire filter graph in preparation for execution of the  
25 processing project, render engine 222 generates a list of sources and when they are

1 required in the filter graph. According to one implementation, the list is referred  
2 to as a reuse list, and is maintained within render engine 222. An example of a  
3 data structure comprising a reuse list is presented with reference to Fig. 41.

4 Turning briefly to Fig. 41, a graphical illustration of an example data  
5 structure comprising a source reuse list is presented. As shown, the reuse list 4100  
6 is comprised of a number of information fields, e.g., 4102-4110 which detail, in  
7 part, the relationship between clips in a track. More particularly, the reuse list  
8 4100 is shown comprising a track identification field 4102, a source identification  
9 field 4104, a project time field 4106 and a source time field 4108.

10 Upon identifying a project source and the associated filters required for pre-  
11 processing the source (i.e., the source chain), render engine 222 assigns each track  
12 an identifier which uniquely identifies the source track within the context of the  
13 filter graph. In this regard, reuse list 4100 includes a field 4102 which maintains a  
14 list of tracks utilized in the associated project. In accordance with the illustrated  
15 example paradigm of the media processing system, the track identifier is utilized  
16 to represent a media clip from a given source.

17 The source identifier field 4104 contains information denoting the project  
18 source associated with a particular track identifier. In this regard, the source  
19 identifier field 4104 may well contain a file name, a file handle, or any other  
20 suitable source identifier.

21 The project time field 4106 denotes at what point during project execution  
22 the media clip is required. The source time field 4108 denotes what portion of the  
23 source file is required to support execution of the processing project. It should be  
24 appreciated that a user may well utilize the whole source file or any part thereof,  
25 as defined by the processing project.

1 In accordance with the illustrated example implementation of Fig. 41, two  
2 tracks are depicted 4110 and 4112. As shown, each of the tracks represent media  
3 from a common source (e.g., source ID 4213) and, the source media clips are  
4 adjacent to one another in the project (e.g., project time 4106) as well as within the  
5 source file (e.g., source time 4108). As will be developed more fully below,  
6 source clips may well be combined in certain situations into a single clip, as is  
7 represented by track 4114 in Fig. 41.

8 Returning to Fig 40 and, in particular, block 4006, render engine 222  
9 reduces the number of source accesses where possible, in accordance with one  
10 embodiment. More particularly, render engine 222 analyzes the reuse list 4100 to  
11 identify opportunities to reduce the number of source accesses by combining  
12 source clips which meet certain criteria. According to one implementation, the  
13 criteria used by render engine 222 include one or more of (1) the source clips must  
14 occur next to one another in the project, (2) the clips appear next to one another in  
15 the source, and (3) the clips must share a common pre-processing source chain  
16 (i.e., must require the same pre-processing). If this criteria is met, render engine  
17 222 may combine the clips into a single clip. More specifically, render engine 222  
18 modifies the reuse list 4100 (Fig. 41) to replace the multiple source accesses  
19 (4110, 4112) with a single source access 4114 representing both source accesses as  
20 a single access. It is to be appreciated that removing a source access improves  
21 filter graph performance and, accordingly, the perceived performance of the  
22 development system by the user.

23 In block 4008, once render engine 222 has reduced the number of source  
24 file accesses (block 4006), render engine 222 dynamically generates and manages  
25 the filter graph to support execution of the development project. In accordance

1 with one aspect, render engine 222 invokes only those source chains associated  
2 with sources that are necessary to support the current and/or impending execution  
3 of the filter graph. It is to be appreciated that by not opening each of the chains of  
4 a processing project, render engine 222 reduces the amount of memory required to  
5 build the filter graph, thereby reducing the amount of memory required to  
6 complete execution of the project.

7 As introduced above, conventional media processing systems may generate  
8 an individual thread each time content was required from a source, even if the  
9 source had been accessed earlier. This redundant loading/unloading of a source is  
10 computationally expensive, and consumes precious memory resources. Extending  
11 the concept of source combining introduced above, a filter and related methods for  
12 sharing a common source and source filter among multiple processing threads will  
13 now be introduced.

14 Turning to **Fig. 44**, a block diagram of an example filter graph 4400 is  
15 presented incorporating a segment filter 4406 which, as will be shown,  
16 dynamically couples a source filter to one or more processing chain. In  
17 accordance with the illustrated example of Fig. 44, filter graph 4400 is depicted  
18 comprising a source 4402, one or more pre-processing transform filters 4404, a  
19 segment filter 4406 and one or more pre-processing transform filter(s) 4408A-N,  
20 each coupled to a matrix switch 308 and rendering filter(s) 4410, 4412,  
21 respectively.

22 As used herein, segment filter 4406 is designed to sit between a source  
23 filter and matrix switch 308 to provide multiple processing chains with source  
24 content from a single source, where it is impossible to combine the source clips (as  
25 introduced above). Render engine 222 invokes an instance of the segment filter

1 4406 after the greatest common pre-processing filter 4404 for each of the chains.  
2 That is, each of the processing chains may require the source content in a different  
3 format (e.g., size, frame rate, decode format, etc.). To the extent that the chains  
4 share common pre-processing attributes, those filter(s) (4404) are placed before  
5 the segment filter 4406 where practicable. In many instances, none of the chains  
6 share common pre-processing and the pre-processing filter(s) merely comprise the  
7 source filter.

8 The segment filter 4406 acts as a controller, or throttle for the source,  
9 instructing the source filter to deliver content from source 4402 at select times.  
10 According to one implementation, the segment filter 4406 is, in turn, controlled by  
11 the render engine 222 and/or the matrix switch filter 308 to provide select content  
12 at select times on select inputs of the matrix switch filter 308. According to one  
13 implementation, the segment filter 4406 issues a “seek” command to the source  
14 filter to request particular content from the source. The source filter then delivers  
15 the requested content through the segment filter 4406 and appropriate pre-  
16 processing filter(s) 4408A-N to deliver the desired content to the requesting matrix  
17 switch 308 to support processing of the development project.

18 As introduced above, render engine 222 is responsive to higher-level user  
19 interfaces, e.g., applications 224. In this regard, it is possible that the filter graph  
20 will receive user-commands while the filter graph is executing the development  
21 project. In accordance with the media processing system paradigm, for example,  
22 it is foreseeable that a user-invoked seek will be received by the filter graph during  
23 execution of the development project. Such user defined commands are typically  
24 serialized with commands issued by filters within the filter graph during the  
25 normal course of execution. In accordance with the illustrated example



1 implementation, where matrix switch 308 “throttles” execution of the filter graph,  
2 matrix switch 308 issues a seek command of its own to the source filter, requesting  
3 the information desired by the user. According to an alternate embodiment, seeks  
4 received from a higher-level application (and, therefore, representative of a user  
5 command) are afforded a higher priority within the filter graph. In such an  
6 implementation, all segment filters 4406 residing within the filter graph are also  
7 notified of such high-priority seeks, so that they can identify what content they  
8 will be required to provide next and, therefore, issue a revised seek command of  
9 their own.

10 The remaining pre-processing transform filter(s) 4408A-N, matrix switch  
11 filter(s) 308 and rendering filter(s) 4410 each function as described in the  
12 application.

13 Turning now to Fig. 45, an example method for generating a filter graph is  
14 presented. More particularly, the method of Fig. 45 is similar to the method of  
15 Fig. 42 wherein render engine 222 attempted source combining of source clips,  
16 which were not project and source time aligned, or which required unique pre-  
17 processing of some sort. In Fig. 42, however, if the source clips were not source  
18 time aligned (4204) and/or the clips required separate pre-processing (block 4206),  
19 each clip was assigned to a separate processing chain. In Fig. 45, however, this  
20 problem is resolved with introduction of a segment filter 4406.

21 More specifically, with reference to Fig. 45, render engine 222 identifies  
22 multiple source clips from a common source which are not source time aligned,  
23 block 4204 and/or require separate pre-processing filter(s), block 4206. Render  
24 engine 222 generates a segment filter 4406 for the filter graph to reuse the source  
25 and at least the source filter, block 4502. That is, the render engine 222 inserts a

1 segment filter 4406 between the source filter and one or more processing chains to  
2 selectively provide otherwise disparate source clips from a single source. But for  
3 use of the segment filter 4406 in the filter graph, the method 4500 of Fig. 45  
4 executes in a fashion similar to Fig. 42, above.

5 Turning now to Fig. 46, a flow chart of an example method of segment  
6 filter operation is presented. In accordance with the illustrated example  
7 embodiment of Fig. 46, the method begins in block 4602, wherein the segment  
8 filter 4406 seeks the source to the place that source data is first needed. As  
9 introduced above, segment filter 4406 receives a request for source content from  
10 matrix switch filter 308. It should be appreciated that insofar as segment filter  
11 4406 may well support a plurality of processing chains coupled to a plurality of  
12 matrix switch filters 308, a number of such requests may be received  
13 simultaneously. According to one implementation, each of the matrix switch  
14 filters 308 assigns a priority to the request for source content, wherein the priority  
15 of the request changes as the time the content is needed draws near. According to  
16 an alternate implementation, render engine 222 determines *a priori* whether source  
17 content will be required simultaneously and, if so, provides a separate source chain  
18 to accommodate such simultaneous content requests, thereby eliminating the  
19 situation of the segment filter 4406 receiving simultaneous requests.

20 In block 4604, the source filter retrieves the requested content and passes  
21 the data to the switch until some sort of indication is received that the end of  
22 content has been received (e.g., an end-of-stream (EOS) indication, an application  
23 interrupt, etc.). As introduced above, an application interrupt may be issued when  
24 a user, through a user interface (e.g., media control application 224), wants to seek  
25 to a certain point in the development project.

1 In block 4606, segment filter 4406 determines whether an EOS or an  
2 application interrupt is received. If not, the process continues with block 4604. If  
3 so, segment filter 4406 identifies the next required segment and when it will be  
4 required, given the current seek location received from the matrix switch filter  
5 308. Based, at least in part, on the current seek location, segment filter 4406  
6 determines whether more segments of the source are required, block 4610. As  
7 introduced above, if a user-defined seek command is issued, it may be issued to a  
8 location in the development project where no further content is required from a  
9 particular source. Thus, segment filter 4406 determines whether additional  
10 segments are required in block 4610.

11 If no further segments are required along one of the processing chains  
12 leading from the segment filter, render engine 222 may remove (at least  
13 temporarily) that chain from the filter graph to free memory space and a matrix  
14 switch filter input for other processing chains, block 4612.

15 If, in block 4610 further segments are required, segment filter 4406 issues a  
16 seek instruction directing the source filter to retrieve and deliver the next segment,  
17 in accordance with the matrix switch filter instructions, block 4614. This process  
18 continues in an iterative fashion with block 4604.

## 20 **The Rejections**

21 **Claim 1** has been amended and recites one or more computer-readable  
22 media embodying a software object for use in a media processing filter graph, the  
23 software object comprising [added language appears in bold italics]:

- 24 • an input, coupled to a media source, to receive content from the  
25 media source; and

- a dynamically determined plurality of outputs, each responsive to the input and coupled to a source processing chain, to provide each of the source processing chains with media content requested from a single instance of the media source in accordance with a user defined media processing project, *wherein said object is configured to reuse the media source by providing disparate source clips from said single instance.*

In making out the rejection of this claim, the Office argues that its subject matter is anticipated by Parry and cites to Parry's column 21, lines 12-25 in support therefore. Applicant has reproduced the content from this excerpt just below:

By way of example only, a filter graph 640, the purpose of which is to play back MPEG-compressed video information from a file may take the form set out in FIG. 20A. Filter graph 640 includes source filter 642, MPEG parser 644, video decompression transform filter 646, audio decompression transform filter 648, video render filter 650 and audio render filter 652. Source filter 642 reads data from a disk and provides it as streaming information to MPEG parser 644. MPEG parser 644 parses the streaming information into its audio and video streams. Transform filters 646 and 648 decompress the video and audio data in the corresponding streams. Render filters 650 and 652 act to display the video data on a screen and send the audio information to a sound card, respectively.

With respect to the recited "dynamically determined plurality of outputs", the Office cites to Parry's parser. Applicant respectfully disagrees that the Parry's parser meets this claim element. Nonetheless, Applicant has clarified the subject matter of this claim to recite that software object "is configured to reuse the media source by providing disparate source clips from [the] single instance" of the media source.

1           The excerpt cited by the Office neither discloses nor suggests any such  
2 subject matter. Accordingly, for at least this reason, this claim is allowable.

3           **Claims 2-14** depend from claim 1 and are allowable as depending from an  
4 allowable base claim. In addition, in making out the rejection of claim 2 under §  
5 103, the Office takes Official Notice that “both the concept and advantages of  
6 alleviating each source processing chain from opening an independent instance of  
7 the source is well known and expected in the art.” Applicant respectfully but  
8 strongly disagrees and respectfully requests that if such is the case, the Office  
9 produce a reference that discloses and teaches the same within the context of this  
10 material’s usage in the claim.

11           **Claim 32** recites one or more computer-readable media embodying a  
12 software object coupled to a source processing chain in a media processing filter  
13 graph comprising:

- 14
- 15       • a software object input, coupled to a media source, to receive content  
from the media source;
- 16       • a dynamically determined plurality of software object outputs, each  
17 responsive to the software object input and coupled to a plurality of  
source processing chain, to provide each of the source processing  
18 chains with media content requested from a single instance of the  
media source in accordance with a user defined media processing  
19 project;
- 20       • the source processing chain comprising:
  - 21       ○ a scalable, dynamically reconfigurable matrix switch having a  
plurality of inputs and a plurality of outputs;
  - 22       ○ at least one matrix switch input being communicatively  
linked with a first processing chain portion;
  - 23       ○ at least one other matrix switch input being communicatively  
linked with a second processing chain portion;
  - 24       ○ the matrix switch being configured to dynamically couple one  
or more of the matrix switch inputs to one or more of the  
25 matrix switch outputs.

1  
2 In making out the rejection of this claim, the Office argues that Parry meets  
3 all of the subject matter of the claim, except for the subject matter that pertains to  
4 the matrix switch. The Office then relies on Hunt and argues that it teaches a  
5 “scalable, dynamically reconfigurable matrix switch...to improve playback of  
6 interactive multimedia content”, citing to column 12, lines 6-58 for support.

7 Applicant respectfully disagrees. The subject matter in Hunt relied on by  
8 the Office simply describes a representation of data in matrix form. For example,  
9 Hunt describes, in column 12 starting at line 18, the assemblage of probability  
10 factors in a table (Table 5) that indicate, in matrix form, the probabilities that  
11 segments will follow one another. Applicant respectfully submits that when Hunt  
12 is considered in its entirety, such data representation has nothing whatsoever to do  
13 with the recited scalable, dynamically reconfigurable matrix switch.

14 Applicant respectfully submits that the Office has misquoted and taken  
15 Hunt out of context. Given this, the Office has failed to establish a *prima facie*  
16 case of obviousness and this claim is allowable.

17 **Claims 33-43** depend from claim 32 and are allowable as depending from  
18 an allowable base claim. In addition, in making out the rejection of claim 37, the  
19 Office takes Official Notice that “both the concept and advantages of alleviating  
20 each source processing chain from opening an independent instance of the source  
21 is well known and expected in the art.” Applicant respectfully but strongly  
22 disagrees and respectfully requests that if such is the case, the Office produce a  
23 reference that discloses and teaches the same within the context of this material’s  
24 usage in the claim.  
25

1           **Claim 44** recites a storage medium comprising executable instructions  
2 which, when executed, implement a system comprising:

- 3
- 4       • means for coupling to a media source to receive content from the  
5       media source to provide an input;
- 6       • means for dynamically determining a plurality of outputs, each  
7       responsive to the input and coupled to a plurality of source  
8       processing chains, to provide each of the source processing chains  
9       with media content requested from a single instance of the media  
10      source in accordance with a user defined media processing project;
- 11      • the source processing chain comprising:
  - 12          ○ a scalable, dynamically reconfigurable matrix switch having a  
13          plurality of inputs and a plurality of outputs;
  - 14          ○ at least one matrix switch input being communicatively  
15          linked with a first processing chain portion;
  - 16          ○ at least one other matrix switch input being communicatively  
17          linked with a second processing chain portion;
  - 18          ○ the matrix switch being configured to dynamically couple one  
19          or more of the matrix switch inputs to one or more of the  
20          matrix switch outputs.

21           In making out the rejection of this claim, the Office makes essentially the  
22 same argument that it did with respect to claim 32. Applicant respectfully submits  
23 that Hunt neither discloses nor suggests a scalable, dynamically reconfigurable  
24 matrix switch as utilized in this claim and described in Applicant's disclosure.  
25 Accordingly, the Office has failed to establish a *prima facie* case of obviousness  
and this claim is allowable.

**Claims 45-48** depend from claim 44 and are allowable as depending from  
an allowable base claim.

1                    **Conclusion**

2                    All of the claims are in condition for allowance. Accordingly, Applicant  
3 requests a Notice of Allowability be issued forthwith. If the Office's next  
4 anticipated action is to be anything other than issuance of a Notice of Allowability,  
5 Applicant respectfully requests a telephone call for the purpose of scheduling an  
6 interview.

7  
8  
9                    Dated: 7/1/04

Respectfully Submitted,

By: 

Lance R. Sadler  
Reg. No. 38,605  
(509) 324-9256